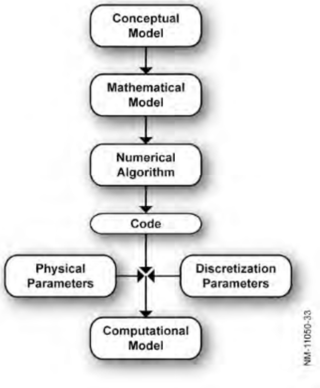
Algorithms and Operation Counting

ASEN 3502 — Computational Methods

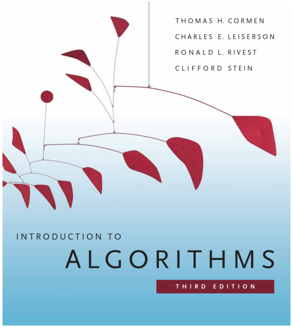


ASME V&V 10, *Guide for Verification and Validation in Computational Solid Mechanics*.

What is an algorithm?

"Informally, an algorithm is any well-defined computational **procedure** that takes some value, or set of values, as **input** and produces some value, or set of values, as **output**."

— Cormen et al.



Introduction to Algorithms
Cormen, Leiserson, Rivest, and Stein

mitpress.mit.edu

Another type of algorithm

Why algorithms?

- Algorithms are mathematical objects. Why are they useful?

Why algorithms?

- Algorithms are mathematical objects. Why are they useful?
  - We can reason about properties like correctness and resource use

Why algorithms?

- Algorithms are mathematical objects. Why are they useful?
  - We can reason about properties like correctness and resource use

    "An algorithm is said to be **correct** if, for every input instance, it halts with the correct output." — Cormen et al.

Why algorithms?

- Algorithms are mathematical objects. Why are they useful?
  - We can reason about properties like correctness and resource use

    > "An algorithm is said to be **correct** if, for every input instance, it halts with the correct output." — Cormen et al.

  - We can make precise and rigorous statements without having to deal with issues like roundoff error, memory allocation, etc.

How do we represent algorithms?

Matrix-vector multiplication: three ways

English

To get entry r of $\{b\}$, multiply each entry in row r of $[A]$ by the matching entry of $\{x\}$, and add up the results. Do this for every row.

Code

```
b = zeros(n,1);
for r = 1:n
    for c = 1:n
        b(r) = b(r) ...
             + A(r,c)*x(c);
    end
end
```

Pseudocode

input : $n \times n$ matrix $[A]$,
vector $\{x\}$
output : $\{b\} = [A]\{x\}$

for $r \leftarrow 1$ **to** n **do**
  $b[r] \leftarrow 0$
  for $c \leftarrow 1$ **to** n **do**
    $b[r] \leftarrow b[r] + A[r,c] \cdot x[c]$
  end
end

"What separates pseudocode from "real" code is that in pseudocode, we employ whatever expressive method is most clear and concise to specify a given algorithm." — Cormen et al.

Analysis of algorithms

- *Analyzing* an algorithm means predicting the resources that an algorithm requires
  - For today, we will count floating point operations ("Flops"): multiplication, division, addition, and subtraction of floating point numbers

Analysis of algorithms

- *Analyzing* an algorithm means predicting the resources that an algorithm requires
  - For today, we will count floating point operations ("Flops"): multiplication, division, addition, and subtraction of floating point numbers

- **Discuss with your neighbor:** Consider the equation $[A]\{x\} = \{b\}$. As n gets larger, which requires more floating point operations? Finding $\{x\}$ given $[A]$ and $\{b\}$, or finding $\{b\}$ given $[A]$ and $\{x\}$?

Matrix-vector multiplication

```
input  : n × n matrix [A], vector {x}
output : {b} = [A]{x}

for r ← 1 to n do
    b[r] ← 0
    for c ← 1 to n do
        b[r] ← b[r] + A[r, c] · x[c]
    end
end
```

Gaussian elimination

input : $n \times n$ matrix $[A]$, right-hand side $\{b\}$
output : solution $\{x\}$ of $[A]\{x\} = \{b\}$

```
// forward elimination
for c ← 1 to n − 1 do
    for r ← c + 1 to n do
        f ← A[r, c] / A[c, c]
        for j ← c to n do
            A[r, j] ← A[r, j] − f · A[c, j]
        end
        b[r] ← b[r] − f · b[c]
    end
end

// back substitution
for r ← n, n − 1, . . . , 1 do
    s ← b[r]
    for j ← r + 1 to n do
        s ← s − A[r, j] · x[j]
    end
    x[r] ← s / A[r, r]
end
```

Asymptotic notation

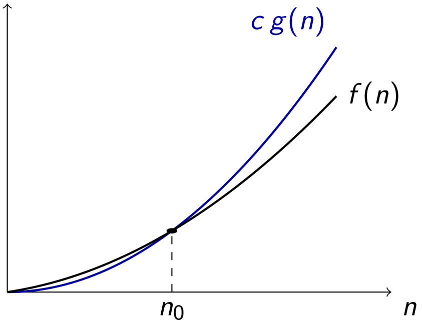
- Asymptotic notation is a tool for keeping track of only the terms that are most important as a function approaches an asymptote

Asymptotic notation

- Asymptotic notation is a tool for keeping track of only the terms that are most important as a function approaches an asymptote
- Chapra: $O(m^n)$ means "terms of order m^n and lower"

Asymptotic notation

- Asymptotic notation is a tool for keeping track of only the terms that are most important as a function approaches an asymptote
- Chapra: $O(m^n)$ means "terms of order m^n and lower"



$f(n) \leq c\,g(n)$ for all $n \geq n_0$

Definition

$f(n) = O(g(n))$ as $n \to \infty$ if there exist positive constants c and n_0 such that

$$0 \leq f(n) \leq c\,g(n) \quad \text{for all } n \geq n_0$$

Gaussian elimination

```
input   : n × n matrix [A], right-hand side {b}
output  : solution {x} of [A]{x} = {b}

// forward elimination
for c ← 1 to n − 1 do
    for r ← c + 1 to n do
        f ← A[r, c] / A[c, c]
        for j ← c to n do
            A[r, j] ← A[r, j] − f · A[c, j]
        end
        b[r] ← b[r] − f · b[c]
    end
end

// back substitution
for r ← n, n − 1, . . . , 1 do
    s ← b[r]
    for j ← r + 1 to n do
        s ← s − A[r, j] · x[j]
    end
    x[r] ← s / A[r, r]
end
```